

# SPECIFICATION

Docket No. 0635MH-40913

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that we, \* and \*, residing in the State of \*, have invented new and useful improvements in a

## **METHOD OF COMBINING SHARED BUFFERS OF CONTINUOUS DIGITAL MEDIA DATA WITH MEDIA DELIVERY SCHEDULING**

of which the following is a specification:

## CROSS REFERENCE TO RELATED APPLICATIONS

The present application claims priority from US Provisional application 60/199,567, filed 4/25/00.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention:

The present invention is in the field of digital communications systems, and more particularly, communications systems which transport continuous digital media such as audio and video.

### 2. Description of the Prior Art:

One of the most important forms of information in digital communications systems is continuous media, exemplified by digital audio and digital video. An example application would be transmission of a movie or live information between a service provider and a user. The traditional method of providing such a service is the *broadcast network*, exemplified by broadcast television or cable television. Models for future information systems use the concept of integrated services, where voice, video and structured information services such as the world-wide web (WWW) are delivered over a single logical transport service, the packet-switched Internet.

To use such a system, the audio and video information must be encoded in digital form, and packetized. To reduce the use of network resources, the

1 information might be first encoded in digital form and then compressed. The  
2 advantage of compression is a significant reduction in data volume, while the  
3 disadvantage is the complexity of the algorithms required for encoding and  
4 decoding the compressed information. For example, directly encoding NTSC  
5 signals from a television system requires about 100 megabits/second of  
6 bandwidth, meaning that a two hour video, at 720 megabytes per minute, would  
7 require about 100 gigabytes of storage. Encoding the signals using the MPEG 2  
8 compression scheme typically reduces the bandwidth required to about 1.5  
9 megabits/second, or 12 megabytes per minute. With such compression an entire  
10 2 hour movie could be stored in about 1.5 gigabytes of storage. MPEG 2 is  
11 designed to require significantly more computation to encode than to decode, as  
12 it is presumed that service providers can afford to perform the expensive  
13 operations once, in order that the inexpensive decoding operations can be  
14 performed many times by receivers. This assumption is clearly true for a stored  
15 video, where the encoding is done once and then decoding is done whenever the  
16 video is viewed.

17       When a system is constructed to distribute digitized continuous media to  
18 many users, there are a number of attractive opportunities for architectural  
19 techniques which can reduce system load in addition to any gains achieved by  
20 operations performed on the media such as efficient coding. In particular, a  
21 powerful technique is *multicast*, where the information is not sent to all possible  
22 recipients, but rather those who indicate interest, perhaps by subscription. To the  
23 degree that sharing can be achieved, e.g., the sharing of a viewing service which

1 plays the video into the network, significant savings can be realized. For  
2 example, rather than sending multiple copies of the same video stream, each at  
3 1.5 megabits/second to a significant population of users, the video might be sent  
4 once via multicast to the set.

5 Multicasts are typically represented as acyclic directed graphs called  
6 *trees*, where the server lies at the root of the tree, there are a set of intermediate  
7 *nodes* interconnected by network *links* which transport information to *leaf nodes*,  
8 at which are located users. A key feature of the intermediate nodes is their ability  
9 to replicate information to several nodes in the direction of the information flow,  
10 so that eventually the information travels from the server to all interested leaves.

11 This basic model assumes that the users are all interested in receiving the  
12 same information at the same time, *e.g.*, a “scheduled” time for a broadcast  
13 event, such as a sports event or a concert. When users join the multicast at  
14 some later time, they may lose the first part of the event (which they may be  
15 interested in) as it is not saved (or “buffered”). For viewing of archived material,  
16 such as replays of videos on demand, the points at which users are interested  
17 and start the viewing will vary sufficiently that multiple time-skewed copies of the  
18 digital continuous media may be being served to users at any given time.

19 The difficulties with this situation are two-fold. First, the server is busy  
20 sending and resending multiple multicast streams of essentially the same  
21 information. Second, the advantages of multicast accrue to the degree that  
22 information is replicated – *unicast*, or point-to-point transmission, can be consider

1 a degenerate case of multicast, and in fact will be the case when there is no  
2 shared interest in a continuous media stream. Thus, to the degree in which we  
3 can aggregate demand for continuous media streams, we can optimize overall  
4 system performance.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
22

## SUMMARY OF THE INVENTION

1           In accordance with the present invention, a communications method utilizes  
2 memory areas to buffer portions of the media streams. These buffer areas are  
3 shared by user applications, with the desirable consequence of reducing  
4 workload for the server system distributing media to the user (client) applications.  
5 The preferred method allows optimal balancing of buffering delays and server  
6 loads, as well as optimal choice of buffer contents for the shared memory buffers.

0635MH-40913 Patent Application

## BRIEF DESCRIPTION OF THE DRAWINGS

1        The novel features believed characteristic of the invention are set forth in the  
2        appended claims. The invention itself however, as well as a preferred mode of use,  
3        further objects and advantages thereof, will best be understood by reference to the  
4        following detailed description of an illustrative embodiment when read in  
5        conjunction with the accompanying drawings, wherein:

6        Figure 1 is a block diagram illustrating multicast over a network;

7        Figure 2 illustrates shared use of a buffer;

8        Figure 3 is a diagram illustrating use of a double buffer; and

9        Figure 4 is a timing diagram illustrating buffer sharing.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

It will be appreciated by those skilled in the art that the following description illustrates techniques that can be utilized in various types of systems. The preferred embodiment is described in the context of an internet system such as is generally known in the art.

### MULTICAST TRAFFIC

In a store-and-forward packet-switched network, multicast is effected by multicasting the packets of data as they arrive at a node in the multicast tree. In practice, the node is a switch, router or other device which receives a multicast packet on an input port, and sends the packet on one or more output ports in accordance with the topology of the graph describing the multicast of the data to users. A multicast from a server to a number of users is illustrated in **Figure 1**.

To perform this operation, the packet must be stored in a buffer memory. In practice, the buffer memory may contain a *queue* of multiple packets. In the continuous media case, a queue containing a sequence of packets represents a segment of the continuous media stream. When the queue consists of large numbers of packets, substantial storage of media can be achieved within the constraints of the node's memory capacity.

While it is often the case that the buffering is mainly used to effect flow control, that is, to mitigate rate differences between senders and receivers, the presence of buffering capability in nodes allows other optimizations where the



1 buffer capacity, *e.g.*, the queued packets, can be shared or reused effectively.  
2 The advantage of sharing and reuse is that once the packet has been transmitted  
3 to a buffer, it need not be transmitted from the server to service a user if the user  
4 can be satisfied with the buffer contents rather than with an additional  
5 independent stream sent by the server.

6 The difficulty in buffer reuse is independent start times for the multicast  
7 streams, as what is in the queue of packets destined to a particular user may not  
8 be the same data as that required by another user. To the degree that work (*e.g.*,  
9 the transmission of data from one node to another) can be avoided through  
10 sharing, the system will perform better. An example of better performance would  
11 be more concurrent users operating with a reduction in bandwidth relative to  
12 similar systems.

### 13 SHARED BUFFERS

14 Each segment of buffered continuous media represents the result of work done  
15 by the network to transport the digitized continuous media to the buffer. Thus, to  
16 the degree that the contents of the buffer can be shared, the work of transporting  
17 the continuous media to the buffer can be shared. An essential observation is  
18 that there is a clear relationship between buffer occupancy and time: each buffer  
19 segment of size **B** bytes represents a playout time of **B/R** seconds, where **R** is  
20 the encoding rate in bytes per second. A related observation was made in John  
21 H. Shaffer, "The Effects of High Speed Networks on Wide Area Distributed  
22 Systems", *Ph.D. Thesis*, CIS, University of Pennsylvania, 1996.

1       The central observation used in our approach is that if additional viewers  
2 arrive within **B/R** seconds of the start time of the original playout, these additional  
3 viewers can utilize the contents of the buffer segment rather than requiring  
4 additional network bandwidth for transmission. To be specific, if the arrival time of  
5 viewers V1 and V2 is separated by less than **B/R**, they can share a buffer  
6 segment. This is illustrated in **Figure 2**.

7       While a related idea was employed in U. Legezda, D. Wetherall and J.  
8 Guttag, "*Active Reliable Multicast*", Proc. Infocomm 1998, SF, CA to reduce  
9 required bandwidth in a reliable multicast, the sharing was different, in that it was  
10 used for recovery of lost multicast packets rather than as a support mechanism  
11 for multicast of digital continuous media. The key basic point of our method is the  
12 use of a per-recipient pointer (an index) into the shared buffer to reflect the  
13 differences in start times. By use of this pointer, the buffer contents can be  
14 effectively shared.

15       There are a variety of parameters which can be adjusted in the design of  
16 such a system. First, to the degree that whole instances of digital continuous  
17 media (for convenience, we will call these "files") can be buffered, there is  
18 significant advantage to be had in that the buffer management algorithms are  
19 simplified. This is because the algorithms need to spend less effort managing the  
20 differences in buffer refresh rate caused by supporting multiple start times. This  
21 management cost is incurred if the whole file is not available, as to save  
22 bandwidth, the buffer contents must be retained until the last viewer is done with  
23 them, *i.e.*, their pointer has advanced to the end of the buffer. This problem is

1 easily addressed if the well-known technique of *double-buffering* is employed to  
2 manage two buffers of size **B** and the start time deltas are limited to **B/R**, as  
3 above. In the double-buffering technique, one buffer is drained by the viewers  
4 while the other is filled from the network, and this solution allows the time-  
5 separated viewers to share the **2\*B** space. The technique is illustrated in **Figure**  
6 **3**.

7 The same effect can be achieved by limiting the time differences between  
8 V1 and V2 to **B/(2\*R)**.

#### 9 OPTIMAL SHARING OF A SINGLE BUFFER

10 There are a variety of techniques for buffer management which can make  
11 use of the buffer capacity to better store continuous digital media. To review  
12 some of the techniques used in US Patent 5,754,938, issued May 19, 1998 to  
13 Herz et al, and US Application No. 09/024,278, filed Feb 17, 1998, both hereby  
14 incorporated by reference hereinto, the buffer contents can be:

- 15 1. Selected based on statistical modeling of the user based on similarity  
16 measures derived from previous viewing.
- 17 2. Can be *prefetched* in advance of viewing demand in order to smooth  
18 demands on bandwidth.
- 19 3. Can be *prefetched* in anticipation if possible viewing demand based on  
20 similarity measures for the viewers sharing the buffer.

1           4. Can be *retained* in anticipation of new viewers requesting the stream  
2           based on similarity measures for users sharing the buffer.

3           While we incorporate by reference the two Herz, *et al.* patents, we wish to  
4           expand slightly here on point 4. This point suggests that the contents of a buffer  
5           should be retained even after all current viewers have viewed the content *if* either  
6           there is no demand for the space it is occupying from other content requests, or if  
7           the likelihood that it will be used again soon is higher than the likelihood that any  
8           prefetched content will be used soon. In effect, the retention decision is one  
9           which takes advantage of the fact that a desirable prefetch is *already in the*  
10          *buffer*. Excepting this observation, the similarity measures and analysis are  
11          included by reference to the other patent and filing.

12          The single buffer case is then optimized by the following algorithm, applied  
13          at each discrete time step in the buffer's existence:

14           1. If the buffer is being used by one or more viewers, examine another  
15           buffer.

16           2. If the buffer has been used recently, estimate the probability that it will  
17           be reused in the near future (e.g., a time interval such as **B/R**). If it is  
18           likely, examine another buffer and mark the buffer "RETAINED".

19           3. If the buffer is marked "EMPTY", and a non-"DOUBLED BUFFERED"  
20           buffer is being used by the maximum number of viewers, fetch the next

# SPECIFICATION

Docket No. 0635MH-40913

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that we, \* and \*, residing in the State of \*, have invented new and useful improvements in a

## **METHOD OF COMBINING SHARED BUFFERS OF CONTINUOUS DIGITAL MEDIA DATA WITH MEDIA DELIVERY SCHEDULING**

of which the following is a specification:

## CROSS REFERENCE TO RELATED APPLICATIONS

The present application claims priority from US Provisional application 60/199,567, filed 4/25/00.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention:

The present invention is in the field of digital communications systems, and more particularly, communications systems which transport continuous digital media such as audio and video.

### 2. Description of the Prior Art:

One of the most important forms of information in digital communications systems is continuous media, exemplified by digital audio and digital video. An example application would be transmission of a movie or live information between a service provider and a user. The traditional method of providing such a service is the *broadcast network*, exemplified by broadcast television or cable television. Models for future information systems use the concept of integrated services, where voice, video and structured information services such as the world-wide web (WWW) are delivered over a single logical transport service, the packet-switched Internet.

To use such a system, the audio and video information must be encoded in digital form, and packetized. To reduce the use of network resources, the

1 information might be first encoded in digital form and then compressed. The  
2 advantage of compression is a significant reduction in data volume, while the  
3 disadvantage is the complexity of the algorithms required for encoding and  
4 decoding the compressed information. For example, directly encoding NTSC  
5 signals from a television system requires about 100 megabits/second of  
6 bandwidth, meaning that a two hour video, at 720 megabytes per minute, would  
7 require about 100 gigabytes of storage. Encoding the signals using the MPEG 2  
8 compression scheme typically reduces the bandwidth required to about 1.5  
9 megabits/second, or 12 megabytes per minute. With such compression an entire  
10 2 hour movie could be stored in about 1.5 gigabytes of storage. MPEG 2 is  
11 designed to require significantly more computation to encode than to decode, as  
12 it is presumed that service providers can afford to perform the expensive  
13 operations once, in order that the inexpensive decoding operations can be  
14 performed many times by receivers. This assumption is clearly true for a stored  
15 video, where the encoding is done once and then decoding is done whenever the  
16 video is viewed.

17 When a system is constructed to distribute digitized continuous media to  
18 many users, there are a number of attractive opportunities for architectural  
19 techniques which can reduce system load in addition to any gains achieved by  
20 operations performed on the media such as efficient coding. In particular, a  
21 powerful technique is *multicast*, where the information is not sent to all possible  
22 recipients, but rather those who indicate interest, perhaps by subscription. To the  
23 degree that sharing can be achieved, e.g., the sharing of a viewing service which

1 plays the video into the network, significant savings can be realized. For  
2 example, rather than sending multiple copies of the same video stream, each at  
3 1.5 megabits/second to a significant population of users, the video might be sent  
4 once via multicast to the set.

5 Multicasts are typically represented as acyclic directed graphs called  
6 *trees*, where the server lies at the root of the tree, there are a set of intermediate  
7 *nodes* interconnected by network *links* which transport information to *leaf nodes*,  
8 at which are located users. A key feature of the intermediate nodes is their ability  
9 to replicate information to several nodes in the direction of the information flow,  
10 so that eventually the information travels from the server to all interested leaves.

11 This basic model assumes that the users are all interested in receiving the  
12 same information at the same time, *e.g.*, a “scheduled” time for a broadcast  
13 event, such as a sports event or a concert. When users join the multicast at  
14 some later time, they may lose the first part of the event (which they may be  
15 interested in) as it is not saved (or “buffered”). For viewing of archived material,  
16 such as replays of videos on demand, the points at which users are interested  
17 and start the viewing will vary sufficiently that multiple time-skewed copies of the  
18 digital continuous media may be being served to users at any given time.

19 The difficulties with this situation are two-fold. First, the server is busy  
20 sending and resending multiple multicast streams of essentially the same  
21 information. Second, the advantages of multicast accrue to the degree that  
22 information is replicated – *unicast*, or point-to-point transmission, can be consider



1 a degenerate case of multicast, and in fact will be the case when there is no  
2 shared interest in a continuous media stream. Thus, to the degree in which we  
3 can aggregate demand for continuous media streams, we can optimize overall  
4 system performance.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2

## SUMMARY OF THE INVENTION

1           In accordance with the present invention, a communications method utilizes  
2   memory areas to buffer portions of the media streams. These buffer areas are  
3   shared by user applications, with the desirable consequence of reducing  
4   workload for the server system distributing media to the user (client) applications.  
5   The preferred method allows optimal balancing of buffering delays and server  
6   loads, as well as optimal choice of buffer contents for the shared memory buffers.

0635MH-40913 Patent Application

## BRIEF DESCRIPTION OF THE DRAWINGS

1        The novel features believed characteristic of the invention are set forth in the  
2        appended claims. The invention itself however, as well as a preferred mode of use,  
3        further objects and advantages thereof, will best be understood by reference to the  
4        following detailed description of an illustrative embodiment when read in  
5        conjunction with the accompanying drawings, wherein:

6        Figure 1 is a block diagram illustrating multicast over a network;

7        Figure 2 illustrates shared use of a buffer;

8        Figure 3 is a diagram illustrating use of a double buffer; and

9        Figure 4 is a timing diagram illustrating buffer sharing.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

1 It will be appreciated by those skilled in the art that the following  
2 description illustrates techniques that can be utilized in various types of systems.  
3 The preferred embodiment is described in the context of an internet system such  
4 as is generally known in the art.

### 5 MULTICAST TRAFFIC

6 In a store-and-forward packet-switched network, multicast is effected by  
7 multicasting the packets of data as they arrive at a node in the multicast tree. In  
8 practice, the node is a switch, router or other device which receives a multicast  
9 packet on an input port, and sends the packet on one or more output ports in  
10 accordance with the topology of the graph describing the multicast of the data to  
11 users. A multicast from a server to a number of users is illustrated in **Figure 1**.

12 To perform this operation, the packet must be stored in a buffer memory.  
13 In practice, the buffer memory may contain a *queue* of multiple packets. In the  
14 continuous media case, a queue containing a sequence of packets represents a  
15 segment of the continuous media stream. When the queue consists of large  
16 numbers of packets, substantial storage of media can be achieved within the  
17 constraints of the node's memory capacity.

18 While it is often the case that the buffering is mainly used to effect flow  
19 control, that is, to mitigate rate differences between senders and receivers, the  
20 presence of buffering capability in nodes allows other optimizations where the

1 buffer capacity, e.g., the queued packets, can be shared or reused effectively.  
2 The advantage of sharing and reuse is that once the packet has been transmitted  
3 to a buffer, it need not be transmitted from the server to service a user if the user  
4 can be satisfied with the buffer contents rather than with an additional  
5 independent stream sent by the server.

6 The difficulty in buffer reuse is independent start times for the multicast  
7 streams, as what is in the queue of packets destined to a particular user may not  
8 be the same data as that required by another user. To the degree that work (e.g.,  
9 the transmission of data from one node to another) can be avoided through  
10 sharing, the system will perform better. An example of better performance would  
11 be more concurrent users operating with a reduction in bandwidth relative to  
12 similar systems.

### 13 SHARED BUFFERS

14 Each segment of buffered continuous media represents the result of work done  
15 by the network to transport the digitized continuous media to the buffer. Thus, to  
16 the degree that the contents of the buffer can be shared, the work of transporting  
17 the continuous media to the buffer can be shared. An essential observation is  
18 that there is a clear relationship between buffer occupancy and time: each buffer  
19 segment of size **B** bytes represents a playout time of **B/R** seconds, where **R** is  
20 the encoding rate in bytes per second. A related observation was made in John  
21 H. Shaffer, "The Effects of High Speed Networks on Wide Area Distributed  
22 Systems", *Ph.D. Thesis*, CIS, University of Pennsylvania, 1996.

1           The central observation used in our approach is that if additional viewers  
2 arrive within **B/R** seconds of the start time of the original payout, these additional  
3 viewers can utilize the contents of the buffer segment rather than requiring  
4 additional network bandwidth for transmission. To be specific, if the arrival time of  
5 viewers V1 and V2 is separated by less than **B/R**, they can share a buffer  
6 segment. This is illustrated in **Figure 2**.

7           While a related idea was employed in U. Legezda, D. Wetherall and J.  
8 Guttag, "*Active Reliable Multicast*", Proc. Infocomm 1998, SF, CA to reduce  
9 required bandwidth in a reliable multicast, the sharing was different, in that it was  
10 used for recovery of lost multicast packets rather than as a support mechanism  
11 for multicast of digital continuous media. The key basic point of our method is the  
12 use of a per-recipient pointer (an index) into the shared buffer to reflect the  
13 differences in start times. By use of this pointer, the buffer contents can be  
14 effectively shared.

15           There are a variety of parameters which can be adjusted in the design of  
16 such a system. First, to the degree that whole instances of digital continuous  
17 media (for convenience, we will call these "files") can be buffered, there is  
18 significant advantage to be had in that the buffer management algorithms are  
19 simplified. This is because the algorithms need to spend less effort managing the  
20 differences in buffer refresh rate caused by supporting multiple start times. This  
21 management cost is incurred if the whole file is not available, as to save  
22 bandwidth, the buffer contents must be retained until the last viewer is done with  
23 them, *i.e.*, their pointer has advanced to the end of the buffer. This problem is

1 easily addressed if the well-known technique of *double-buffering* is employed to  
2 manage two buffers of size **B** and the start time deltas are limited to **B/R**, as  
3 above. In the double-buffering technique, one buffer is drained by the viewers  
4 while the other is filled from the network, and this solution allows the time-  
5 separated viewers to share the **2\*B** space. The technique is illustrated in **Figure**  
6 **3**.

7 The same effect can be achieved by limiting the time differences between  
8 V1 and V2 to **B/(2\*R)**.

### 9 OPTIMAL SHARING OF A SINGLE BUFFER

10 There are a variety of techniques for buffer management which can make  
11 use of the buffer capacity to better store continuous digital media. To review  
12 some of the techniques used in US Patent 5,754,938, issued May 19, 1998 to  
13 Herz et al, and US Application No. 09/024,278, filed Feb 17, 1998, both hereby  
14 incorporated by reference hereinto, the buffer contents can be:

- 15 1. Selected based on statistical modeling of the user based on similarity  
16 measures derived from previous viewing.
- 17 2. Can be *prefetched* in advance of viewing demand in order to smooth  
18 demands on bandwidth.
- 19 3. Can be *prefetched* in anticipation of possible viewing demand based on  
20 similarity measures for the viewers sharing the buffer.

1           4. Can be *retained* in anticipation of new viewers requesting the stream  
2           based on similarity measures for users sharing the buffer.

3           While we incorporate by reference the two Herz, *et al.* patents, we wish to  
4           expand slightly here on point 4. This point suggests that the contents of a buffer  
5           should be retained even after all current viewers have viewed the content *if* either  
6           there is no demand for the space it is occupying from other content requests, or if  
7           the likelihood that it will be used again soon is higher than the likelihood that any  
8           prefetched content will be used soon. In effect, the retention decision is one  
9           which takes advantage of the fact that a desirable prefetch is *already in the*  
10          *buffer*. Excepting this observation, the similarity measures and analysis are  
11          included by reference to the other patent and filing.

12          The single buffer case is then optimized by the following algorithm, applied  
13          at each discrete time step in the buffer's existence:

14           1. If the buffer is being used by one or more viewers, examine another  
15           buffer.

16           2. If the buffer has been used recently, estimate the probability that it will  
17           be reused in the near future (e.g., a time interval such as **B/R**). If it is  
18           likely, examine another buffer and mark the buffer "RETAINED".

19           3. If the buffer is marked "EMPTY", and a non-"DOUBLED BUFFERED"  
20           buffer is being used by the maximum number of viewers, fetch the next



**B** bytes of the continuous media stream into the new buffer to achieve double buffering and mark both buffers "DOUBLE BUFFERED".

4. If the buffer is marked "EMPTY" and similarity measures show that prefetching more of the stream would optimize performance of the users of the buffer, mark the buffer "PREFETCHING" and request that a continuous media stream of **B/R** duration be sent to fill the buffer.

5. If a buffer is PREFETCHING and a buffer is required for on-demand traffic, mark the buffer "EMPTY".

6. If the buffer has been used, and it is unlikely to be reused in the near future, mark it "EMPTY".

7. If the buffer has not been used, mark it "EMPTY".

An interesting consequence of this algorithm is that highly popular data will be either completely prefetched or after prefetching and/or viewing, wholly retained. Thus, without explicitly requiring whole file caching, the system will naturally achieve it, and will do so on the basis of statistical usage and estimation measures.

There are additional gains to be made because streams are generally served over relatively lengthy periods of time and are consumed sequentially. This is especially true for such data types as audio and video programming. Because the full stream is delivered to a particular point, it is possible to predict content demand quite precisely for the following minutes or even hours.

1 Moreover, when geographically proximate nodes are conveying the same  
2 stream, even at a time shift, it is possible for them to economize on upstream  
3 bandwidth.

4 For example, if node N1 in the network is relaying frames from the FIRST  
5 hour of Citizen Kane to one or more downstream users, then the probability is  
6 extremely high that it will eventually have to relay frames from the SECOND hour  
7 in Citizen Kane, and we can accurately estimate when it will have to do so.

8 If some nearby node N2 happens to have such frames in a buffer (e.g., N2  
9 is already relaying the second hour), then it is cheap for N1 to get those frames.  
10 This cheapness is temporary: N2 isn't going to keep the frames forever, so it may  
11 make sense for N1 to immediately copy N2's frames – that is, to prefetch them.

12 In other words, N1 is getting its stream from the head end, but it is also  
13 monitoring N2's stream and caching it for later. Eventually N1 will be able to  
14 switch to the cache, at which point it can drop the head end.

15 Note that the streams don't need to be transmitted at the same speed at  
16 which the user will consume them – with additional bandwidth, it would be  
17 possible, for example, to send a node the remaining head end of a video stream  
18 in a rapid burst. If multiple streams are being fed to a localized cluster of nodes,  
19 such a bursting strategy would allow the many streams to be rapidly collapsed  
20 into very few (if not one) shared streams, greatly decreasing bandwidth use by  
21 the locale.

1 More complex probabilistic strategies are also possible, with each node  
2 participating in a self-organized market, requesting and taking bids for slices of  
3 bandwidth.

#### 4 OPTIMAL SYSTEM DESIGN WITH BUFFERS AND SERVERS

5 It is obvious that the scheme can be applied to a server with a single  
6 buffer. When a more complex system, such as that in **Figure 1** is constructed,  
7 the buffers can be viewed as a shared resource and as multiple layers of  
8 intermediate capability. In such a system, we can use a hierarchical scheme, so  
9 that individual viewers using a single buffer are replaced by multiple viewers  
10 sharing a buffer, which in turn shares buffers in the multicast tree. Further, the  
11 buffers can cooperate amongst themselves to share caching capacity, and  
12 further, the similarity measures of the patents included by reference can be  
13 localized to the population of viewers near the particular buffer. In such a case,  
14 the population statistics are localized. Caches that are higher in the tree (nearer  
15 to the server) aggregate more and more traffic, but have better models of user  
16 demand from the aggregated demand of the buffers which are presenting  
17 aggregated demand to them.

18 Additionally, multiple servers can be used. In this case, buffers may  
19 communicate with multiple servers based on the demands of their users, and  
20 prefetch information based both on user interest and aggregated interest, and  
21 capacity of the server being used.

1 The advantage of these architectural advances for scaling is significant.  
2 Information in caches near the edges of the system is localized to the user  
3 interests represented by the similarity measures applied to the continuous digital  
4 media or descriptive information associated with it. Each level of buffering in the  
5 hierarchical multicast demand aggregates various levels of interest and optimism  
6 in prefetching data: data prefetched by many buffer caches on behalf of their  
7 clients/viewers will be more likely to be buffer resident where higher levels of  
8 aggregation ensue.

#### 9 VI Demand Aggregation of Data Streams as an Optimally Bandwidth Conserving 10 Form of Pre-Fetching

11 The following technical methodology describing similarity-informed pre-  
12 fetching (the subject of a co-pending patent application by the inventors of the  
13 presently disclosed invention) provides an underlying technical framework for the  
14 object of the novel innovative concept introduced within this section which is the  
15 aggregation of data streams based upon capitalizing upon aggregate demand  
16 prediction of a sub-population of users who are imminently likely to request a  
17 particular file (a standard file or a streaming file). Similarity-informed pre-fetching  
18 provides a fundamental technical basis for demand aggregation of down-loaded  
19 data streams with a few fundamental differences as are explained further below.

20 Bandwidth is greater at the root end of a hierarchical node network  
21 system, in anticipation of a request, it is prudent to use the similarity measure to  
22 predictively cache files into local servers and further to narrow-cast selections

1 into given distribution cells and (hierarchical) sub-cells through sub-servers  
2 based on what selections are most likely to be requested in each cell so as to  
3 significantly increase the utilization of bandwidth via this hierarchical narrow-cast  
4 configuration. The importance of this savings appears in proportion to the degree  
5 of granularity (smallness of the cell) in the narrow-cast architecture. This  
6 technique can also be used to make decisions for scheduling what data should  
7 be placed on dedicated channels. This may be more network-efficient if a file  
8 were popular enough to be continuously in the queue because upon submission  
9 of a request a file may be partially downloaded regardless of where in the length  
10 of the file the download began. The (initially) missed portion of the file can then  
11 be immediately picked up as the file narrow-cast starts over, thus completing the  
12 download in the same time period as if a special request for the file were made.  
13 Mobile users whose geographic locations are known can have files pre-cached  
14 (e.g. at night) into the servers which are presently physically closest to them at  
15 any given time.

16 The present data distribution system employs the idea of pre-fetching,  
17 which has also been referred to as pre-caching, cache-pre-loading, or  
18 anticipation in the technical literature. The basic idea is that if good predictions of  
19 future data requirements are available, and there is excess data-fetching  
20 capability available, the data should be fetched aggressively in anticipation of  
21 future needs. If successful, this technique has two major benefits applicable to  
22 present and future networks. First, it can reduce (i.e., improve) response-time, a  
23 major performance advantage in interactive systems. Second, it can reduce

1 congestion and other problems associated with network overload. To  
2 understand how the responsiveness of the system is improved, the unused  
3 bandwidth can be used to transmit information likely to be used in the future. For  
4 example, if a list is being traversed 1,2,3,4, it is likely that if object N has been  
5 requested, that object N+1 will be the next request. If N+1 is pre-fetched from  
6 the remote system, it will be available when the request is made with additional  
7 delays. All of the "UNUSED BANDWIDTH" can potentially be used to pre-fetch.

8 Within the context of the pending patent on the pre-fetching concept  
9 entitled "Broadcast Data Distribution System with Asymmetric Uplink/Downlink  
10 Bandwidths" one of the key objectives is it involves the use of pre-fetching as a  
11 congestion control technique as if we pre-fetch successfully during more lightly  
12 loaded periods (such as TIME=0.42), we reduce the probability of data being  
13 requested in the futures, essentially trading the guarantee of a fully loaded  
14 network today for the promise of no congestion in the future. By fetching data in  
15 anticipation of future needs, we reduce (at least probabilistically) those future  
16 needs.

17 Pre-fetching has been used in the computer operating systems field for  
18 several decades, and a variety of algorithms have been explored. A. J. Smith of  
19 Berkeley has reported that the only case where successful predictions about  
20 future requests for memory objects can be made is when accesses are  
21 sequential. More recent work for higher-level content such as World-Wide Web  
22 (WWW) hypertext has shown that user-authored links to other hypermedia  
23 documents can be followed with some success.

1 The pre-fetching technology which the inventors of this disclosure have  
2 previously invented is based on unused slots being filled with pre-sent  
3 information based on our understanding of user interest using the similarity  
4 measures developed in the issued patent entitled "System for Generation of User  
5 Profiles for a System for Customized Electronic Identification of Desirable  
6 Objects" (US Patent 5,754,939) and used for our prioritization (see p. 18 of the  
7 above-referenced invention disclosure), a concept that they do not deal with, as  
8 they follow http: links based on observations about the high probability that these  
9 links will be followed by users.

10 The pre-fetching invention may be usefully applied within the context of  
11 set-top box like devices such as personal digital assistants or network  
12 computers, or personal computers used as a form of set-top box as well as any  
13 type of "fat client" as a method of reducing response time as observed by users.  
14 This method used "links" to other documents embedded in an HTTP-format file  
15 as hints that those links should be followed in pre-fetching data; that is, the linked  
16 documents should be fetched in anticipation of the user's desire to follow the  
17 links to those documents.

18 The present invention provides two enhancements to this scheme. First, it  
19 provides a technological means by which the pre-fetched data can be intermixed  
20 with on-demand data to provide overall improvements in response time to a large  
21 population of HTTP/WWW users, with reduced memory requirements. Second,  
22 the present invention, which views the down-link as a fixed capacity resource,  
23 provides a general scheduling method embodying techniques such as user

1 preferences to pre-fetch when slots or bandwidth are underutilized, to  
2 preemptively reduce future demand for bandwidth.

3 In addition, the similarity measures which suggest:

- 4 1. An anticipated behavioral similarity between different though metrically  
5 "similar" users may be used to further analyze other previous user's  
6 on-line behavior as well as,
- 7 2. Page similarity to other links, which the user has an explicitly or  
8 implicitly (predicted) affinity towards, may be a technique to further  
9 improve predictive accuracy as to which of the available links the user  
10 is probabilistically most likely to imminently select next (compare to  
11 aggregatively using the overall popularity of those links).

12 The general technique of using similarity-informed pre-fetching is  
13 described at length in US patent entitled "Pseudonymous Server For System For  
14 Customized Electronic Identification Of Desirable Objects", U.S. Patent No.  
15 5,754,938 filed October 31, 1995, issued May 19, 1998.

16 The technique of data-streams is based upon a similar variation of the  
17 type of similarity-informed pre-fetching, which is performed on a relatively  
18 dynamic basis as suggested below. A couple of fundamental extensions to the  
19 basic technique are added, however. Unlike dynamic similarity-based pre-  
20 fetching, because there is a certain degree of predictive error, which invariably  
21 occurs though the present techniques attempt to anticipate imminently



1 forthcoming file requests on the part of the user (and where the error rate  
2 exponentially increases in proportion to the length of the anticipatory period), the  
3 basic goal of this dynamic pre-fetching is increasing speed of user access to  
4 page requests at the expense of the additional bandwidth, which is consumed as  
5 a result. However, in the case of data stream aggregation the key objective is to  
6 minimize bandwidth utilization. This approach also results in the ability to not  
7 adversely affect speed of access to the data. Although its use is in no way  
8 limited, it is likely that the present approach to pre-fetching basic data stream  
9 aggregation may be particularly well suited for delivering data to the leaf end  
10 nodes of the network

11 Compared to similarity-based pre-fetching, the key modification in the  
12 basic method (and certainly technical challenge) of pre-fetching based data  
13 stream aggregation is the following:

14 In similarity-based dynamic pre-fetching (as suggested above), the  
15 similarity measurements of the predictive data model anticipates, on a dynamic  
16 basis, the forthcoming file request actions of the user. In fact the outputs, i.e.,  
17 probabilities of a given user requesting any given file, can be measured as a  
18 function of time (T). If this measurement is based upon the entire subset of that  
19 user population which has at least some non-zero probability of imminently  
20 selecting that file, we, in turn, may determine the aggregate probability of that file  
21 to be requested by the entire user sub-population (i.e., which is serviced by a  
22 particular data distribution server) as a function of time. Of course, in  
23 accordance with the probabilistic model, this aggregate probability curve (of

1 average likelihood of the user population to initiate the request) changes on a  
2 moment to moment basis changes in accordance with further behavioral  
3 feedback of the sub-population as time (t) approaches the average  
4 (probabilistically most likely) time for overall demand for that file to culminate  
5 (however, this value remains fixed for our purposes as once a pre-fetching  
6 decision has been made any subsequent probability shifts are irrelevant). The  
7 point in time at which a data stream is scheduled to initiate is  $T_1$  and the end is  
8  $T_2$ . There is another important relationship which is within a given average  
9 probability measure the actual time from the point in time which is that moment  
10 that the user population on average is anticipated to actually request the file. We  
11 will call this value  $T_{b1}$ . The end of that period is called  $T_{b2}$ . We want to also  
12 determine the average effective available memory of each relevant user's client's  
13 local buffer, however, this value is also affected by such variables which compete  
14 for this space such as how much of this memory had been pre-allowed to long-  
15 term pre-fetching (the relative proportions of such allocation of which would be  
16 determined through experimentation and may be network specific) and how  
17 much "risk diversification" is provided for i.e., the probability distribution for any  
18 given individual at any give instant in time (t) preceeding an actual request is very  
19 likely to contain secondarily another (or other) file(s) with some non-zero  
20 probability. Thus the total probability (and possibly relative probability)  
21 determines available buffer memory for the present anticipated file request. For  
22 purposes of the present estimation, this value is called  $\Delta T_b$  and is measured as  
23 the amount of time that the present effective available memory buffer for that

1 client is able to receive the data stream associated with the present anticipated  
2 forthcoming request.

3 We then want to select a time T to actually request the file for delivery to  
4 all the relevant users U based upon the average of the product of probability (of  
5 making the desired relevant file request) ( $P_b$ ) and time such that  $T_1$ 's value is  
6 such that period.

7  $T_1 - T_2$  provides the maximum possible product of probability and time for  
8 all relevant clients' buffers collectively (that are able to concurrently co-exist  
9 within period  $T_1 - T_2$ ). (the fixed value describing the period of the data stream)  
10 Based upon these variables the key technical challenge of pre-fetching-based  
11 data stream aggregation is determining values of  $T_1 - T_2$  which achieve  
12 optimality in reducing bandwidth consumption in the multi-cast of that particular  
13 data stream. This is represented in Fig. 4 by our attempt to find a  $T_1$  value  
14 optimizing the area described by probability and time within the  $T_1 - T_2$  period.

15 The following equation is provided:

16 
$$(T_1 - T_2) = \{\Delta T_b P_b Z + (\Delta T_b + 1) (P_b + 1)(Z + 1) + \dots\}(\text{mean average})$$

17 where Z is the percentage of  $T_{b1} - T_{b2}$  which overlaps with  $T_1 - T_2$ .

18 Fig. 4 depicts graphically how a key objective of the above equation is to  
19 find a  $T_1 - T_2$  value which maximizes the (2 D) area under the various client  
20 buffers collectively in a probability time graph.

1 Finally, in light of the present technique because the predictive models are  
2 prone to a certain degree of oversight, i.e., not anticipating the forthcoming  
3 request actions certain uses for a given file or not the accurate timing therefore  
4 (e.g., not anticipating the request action soon enough) there will invariably exist  
5 certain inefficiencies in the present model in which certain user request data  
6 streams are not properly (or not at all) aggregated, thus, we would like to provide  
7 yet another additional solution to more efficiently aggregate these inefficiently  
8 transmitted streams. The idea is that if a local client buffer has failed to initiate  
9 download of a stream (or has initiated it after its initialization, we can effectively  
10 re-transmit the "missed" portion of the stream to the new requester at a very fast  
11 rate (e.g., if it is streaming media content, considerably faster than real-time) up  
12 until the point in which the data received by the new requester "catches up" with  
13 the original stream at which time both the new requester and original requester(s)  
14 then share the same stream for the remainder of its duration. In such an event  
15 the bandwidth utilization specifically allocated to that missed portion of the file we  
16 could say becomes "bursty". Due to the somewhat different mechanism of both  
17 data stream aggregation methods, in trying to achieve optimal bandwidth  
18 utilization the degree to which this additional approach to aggregation should be  
19 relied upon compared to the original pre-fetching based approach (that is to say  
20 from a probabilistic standpoint how speculatively do we want to pre-fetch versus  
21 rely upon effectively the present ("fall-back") approach of using burstiness of  
22 transmission to compensate for the resulting "missed" requests of the predictive  
23 (anticipatory) approach. This balance between the degree of utilization of these

1 two methods may also be a bit subjective in as much as if bandwidth is (presently)  
2 overly constrained, the system may automatically adjust by relying more heavily  
3 upon the original technique of pre-fetching based data stream aggregation (and  
4 certainly even more towards artificial delays in as much as bursty data stream  
5 aggregation while involving no delays in initiating does involve some degree of  
6 extra bandwidth utilization during the "bursts". Of course, as part of this pre-  
7 fetching procedure and (relatedly) the ) the number of files provided a given  
8 probability distribution for forthcoming requests of those files must also be  
9 determined through further experimentation.

10 In another form of data stream aggregation called "artificial delays",  
11 described further below, speed of access to the user is invariably compromised  
12 in direct proportion to the degree of data stream aggregation which is desired for  
13 reducing bandwidth. In the presently described version of data stream  
14 aggregation (as is also true in the case of artificial delays), although it may be  
15 variable, on the average, the number of users serviced by the data distribution  
16 server as well as the degree of popularity of the particular file being presently  
17 requested by the user relative to that particular population of users, is directly  
18 proportional to the speed of access by the user to that particular requested file.  
19 Accordingly, non-dynamic pre-fetching (also detailed above), because it is both  
20 non-dynamic in nature and also achieves a reduction in overall network traffic  
21 can be used in combination with the present pre-fetching based data stream  
22 aggregation in order to provide an optimally efficient traffic reduced network  
23 environment, and as is further described below there are further bandwidth

1 reduction optimization techniques which allow for the approach between the  
2 intermediate node and leaf end of the network while allowing for non-dynamic  
3 similarity-informed pre-caching and the most bandwidth efficient data stream  
4 aggregation technique, i. e., artificial delays to occur on nodes close to "trunk"  
5 portion of the network where the potential number of user connections  
6 represented is extremely large and thus potential for bandwidth conservation,  
7 using artificial delays is extremely great. On the other hand, if near the leaf end  
8 of the network further bandwidth conservation is desired, it is even possible to  
9 provide the technique of artificial delays in combination with static pre-caching  
10 and dynamic pre-fetching based data stream aggregation. And this approach  
11 may be particularly compelling in achieving substantial bandwidth conservation if  
12 the end-user population (or number of network "leaves") is quite large.

### 13 Artificial Delays

14 This note discusses the notion of "artificial delays" in the queuing of  
15 requests for the satellite or cable system to which our set top boxes are attached.  
16 The idea is that by careful management of the queues, we can effect significant  
17 bandwidth savings for the system as a whole. If you will remember, the Server  
18 scheduling algorithm (I've attached the text for it from the DBS scheme I sent  
19 during the Summer at the end of this e-mail) goes like this:

20 The client set-top box (of which there should be many) sends REQUESTs  
21 for information in cell-sized units to the server system. The server system applies  
22 a priority algorithm (see especially Step 10, below) to CHOOSE the next cell to

1 send. By design of the relative priorities, we can get good responsiveness and  
2 reduced bandwidth needs, in spite of the low memory needs (and low cost) of the  
3 set top boxes.

4 Imagine the scenario where there are MANY set tops connected to the  
5 server. This situation might be where the Cs are Clients and the S is a Server.  
6 Now clearly, there is a multiplicity of Clients, and by virtue of this multiplicity, we  
7 may be able to achieve a savings through appropriate delays. I again believe the  
8 similarity measure is the key to success here, and to novelty. Consider the cell  
9 requests for Clients C1, C4 and C5, shown below using letters to indicate  
10 particular cells as discussed in our disclosure text.

11 C1: E-T-A-O-I-N-S-H-R-D-L-U.....

12 C4: N-A-T-I-O-N-A-L-V-E-L.....

13 C5: E-A-T-O-N-L-Y-S-U-D.....

14 We mark these cell request with times associated with their transmission  
15 intervals:

16 T:

17 I:0 0 0 0 0 0 0 0 1 1 1 1

18 M:1-2-3-4-5-6-7-8-9-0-1-2-3.....

19 E:

20 Now, for convenience, assume that all of the cell requests show above  
21 have the same priority. Then the server might actually send the following  
22 sequence of cells over the channel:

1                   S:E-N-E-T-A-A-A-T-T-O-I-O.....

2           Thus, we are servicing the cell requests C1-C4-C5-C1-C4-C5..... (in fact,  
3 the Server may notice the overlaps between requests by C1 and C5 in the first  
4 interval, C4 and C5 in the second interval, C4 and C5 in the third interval, and C1  
5 and C5 in the fourth interval, giving:

6                   S:E-N-T-A-A-T-O-I-I-O-N-N...

7                   )

8           Imagine that the clients are always listening. Then, we can delay cell  
9 requests in the HOPE that the REPLYs can be MERGED, satisfying multiple set-  
10 top-box clients with the same REPLY. To make this concrete, consider delaying  
11 service by one period. So the output of the server then looks like:

12                   S:#-E-N-T-A-O-I-N-L-S-A-Y....

13           What is going on here is very subtle. By delaying some clients service  
14 requests, we are INCREASING THE PROBABILITY that another such request  
15 will come in, which can we can fold into service of the equivalent delayed  
16 request. The cost is potentially in delay, but with enough overlap, the cell times  
17 are short enough for 48 bytes on a DBS channel that we can probably delay  
18 significantly.

19           Considering the problem theoretically for a moment, we can compute the  
20 gain for the an acceptable delay D as being the number of redundant  
21 transmissions which are eliminated due to a delay D. So, for delays of from 1 to



1 10 cell times the total number of DBS cells without redundancy checks is 30; the  
 2 number required when this small optimization is applied is as shown below:

3 Delay DBS CellsBandwidth Savings

4 -----

5  $02430/24=25\%$

6  $11830/18=66\%$

7  $21730/17=76\%$

8  $31730/17=76\%$

9  $41530/15=100\%$

10  $51530/15=100\%$

11  $61530/15=100\%$

12  $71530/15=100\%$

13  $81530/15=100\%$

14  $91530/15=100\%$

15  $101430/14=114\%$

16 We compute the bandwidth gain against the dumb use of 30 cell times;  
 17 the bandwidth gain comes from the fact that the synchronous satellite channel  
 18 gives us a fixed bandwidth, giving a fixed number of cells per unit time, and we  
 19 have just saved 16 cell times by use of the delay scheme. For this example, at  
 20 this point no more gain is possible, since all the duplication has been eliminated.  
 21 In some sense, this behaves like a compression scheme. The similarity  
 22 algorithm increases the probability that these overlaps will occur - the ideal  
 23 situation is we are waiting long enough so that the scheduled broadcast cell

1 satisfies almost all requests for that cell within a significant time intervals (say  
2 several milliseconds).

### 3 Optimal Stream-Handling Techniques as a Function of Node Location in the 4 Network.

5 The method by which streams are handled depends in part on a node's  
6 location in the network.

7 A node close to the network's "trunk" will often be best served by artificial  
8 delays and pre-caching techniques, whereas nodes closer to the network's  
9 "leaves" more efficiently make use of demand aggregation and pre-fetching.

10 The particular topology of a given network will determine the distance a  
11 node must be from the center in order for the "leaf" approaches to be more  
12 appropriate than the "trunk" approaches.

### 13 Long-Term vs. Short-Term Pre-Caching –

14 Due to the relative importance of responsiveness compared to relative  
15 bandwidth savings opportunities at the edges (leaf ends) of the network compared  
16 to the trunk , dynamic pre-fetching (and pre-fetching-based data stream  
17 aggregation) may be efficiently utilized near the network edges while more static  
18 (long-term) pre-caching )may be more appropriate closer to the trunk nodes of the  
19 network. Also at this trunk level because the number of repetitive requests artificial  
20 delay based data aggregation is very efficient and because of the rather large

1 storage capacity associated with this repetitive file traffic, long-term pre-caching is  
2 certainly ideal.

### 3 Automatic Adaptive Shifting to Other Techniques –

4 E. g., if/when traffic congestion and slow down at the edges occurs, it may  
5 be prudent to shift the relative degree of utilization of one technique to another ,  
6 e.g., in the present scenario the use of artificial delays may be able to  
7 significantly reduce delays through relieving congestion. Pre-fetching-based data  
8 stream aggregation may also provide useful advantages.

9 As a result, different levels in the network have different traffic and storage  
10 characteristics (and relative responsiveness priorities). The selection and relative  
11 dependence upon one of the above techniques versus another may be different  
12 at these different levels and the desirable optimizations requiring different uses  
13 and priorities of these various techniques may change at any given level in the  
14 network and at any given time based upon these dynamically changing  
15 characteristics of the network traffic (to the extent that these dynamic  
16 characteristic changes are not pre-determinable using existing network traffic  
17 intelligence solutions).

18 Likewise, because both probability and statistical confidence (degree of  
19 certainty) varies even between different files which are predictively anticipated  
20 (using pre-fetching-based data stream aggregation), the use of this method (the  
21 degree of reliance) of this method, may vary even between anticipated file  
22 requestion, e.g., it may be advantageous if the probability of a file being requested

1 is low or the statistical confidence of a file (even the most “likely” file) to make the  
2 anticipatory period for requesting that file low (i.e., less speculative) in order to rely  
3 more upon bursty data stream aggregation (or artificial delays particularly in these  
4 latter two techniques the period of anticipation may be ideally further minimized  
5 if/when these other techniques are utilized with higher relative importance.

#### 6 Consideration for Pre-Fetching Based Data Stream Aggregation –

7 In addition, because data stream pre-fetching aggregation is a very important  
8 technique for providing bandwidth conservation (an important need for the leaf  
9 ends of the network), while also providing a high degree of responsiveness to  
10 requests, it may also be important to insure that the appropriate technique (or  
11 combination of techniques) as suggested above is utilized to insure that files  
12 reach this bandwidth bottleneck in the network, (e.g., the intermediate node just  
13 prior to leaf end node) such that pre-fetching data stream aggregation at this  
14 level can occur extremely efficiently without unnecessary efficiency  
15 consequences resulting from delays higher up in the network (e. g., this may be  
16 especially true in the case of the wireless network example).

### 17 ADDITIONAL CONSIDERATIONS

#### 18 1. Streaming vs. Static Files

19 This description has been focused mainly on streaming types of data.  
20 However, packet analysis at the backbone level could identify metrically close  
21 data transmissions of any sort (that is, streaming or static (“static” meaning non-

streaming). Thus, when two proximate nodes download the same extremely large data file of a non-streaming sort, they could use a shared buffer to reduce their external need of bandwidth to a single connection. In the end, this architecture handles files of both the streaming and static types in very much the same way.

## 2. Tradeoffs between Bandwidth and Memory

In all of these examples, there is a continuous tradeoff between bandwidth and memory: if a great deal of bandwidth is available, there is little need for localized buffers – every user can afford an individual real-time connection and therefore has no need to store any streams. On the other hand, if a great deal of memory is available, the permanent storage of all incoming streams to a local server would eventually reduce the need for external bandwidth.

## EXEMPLARY APPLICATIONS

The network architecture described in this patent can be embodied in many relevant applications. Note that although a few are described here, the architecture is generally applicable to any situation in which multiple proximate (relative to the network) users will potentially access the same stream of data within some period of time.

### 1. Neighborhood Server

A subdivision of homes is linked by high-bandwidth lines to a shared, local server which acts as the neighborhood's central portal to the Internet. Using

1 residents' video preference profiles, the server predicts which movies or  
2 television programs are most likely to be requested as downloads, and buffers  
3 (for example) the first ½ hour of the 50 most likely selections to local memory.  
4 This could best be done during the day, when residents are more likely to be at  
5 work, temporally smoothing the neighborhood's demand for bandwidth.

6 In the evening, if a resident happens to select a video stream which has  
7 already been front-loaded in the local server, he is sent data directly from the  
8 local buffer. If no other resident requests the same program, the single user is  
9 switched to an external stream when the buffer has been exhausted. However, if  
10 several residents select the same program at roughly the same time (that is,  
11 within the ½ hour buffer), the server continuously downloads (and buffers) a  
12 window of streamed data that spans the users' timings. For example, Resident A  
13 starts watching Citizen Kane; for the first ½ hour he is initially fed a stream from  
14 the local server. Suppose 5 minutes later Resident B also starts to watch Citizen  
15 Kane. When A reaches the end of the ½ hour buffer, the neighborhood server  
16 starts to draw Citizen Kane as a stream directly from the internet, pushing it to A  
17 directly, and then saving the stream to a continuously-refreshed five-minute  
18 buffer. B is fed from the end of the buffer, just before the stream is finally cleared  
19 from memory. In this way, rather than opening two high-bandwidth connections  
20 to the Internet, the local neighborhood server needs only open a single high-  
21 bandwidth connection and allocate enough memory in its local buffer to hold five  
22 minutes of video programming. When multiple residents watch the same

1 programming at fairly similar times, this method greatly reduces the subdivision's  
2 overall need for external bandwidth.

3 Note too that peer-to-peer (P2P) methods could be used to expand the  
4 neighborhood's available storage – the neighborhood server would be given  
5 permission by residents to temporarily make use of memory or hard disk space  
6 that they are not currently using on their own home machines. This would  
7 expand the number of stream front-ends that could be prefetched.

## 8 2. Demand Aggregation for Wireless Electronics

9 Demand aggregation techniques would also be useful in the case of  
10 wireless devices -- if many users in a particular locale exhibit similar data needs  
11 (such as Wall Street executives checking popular stock prices periodically),  
12 bandwidth could be conserved by continuously sending the information in a  
13 single stream to a server connected to the local wireless transmitter.

## 14 3. Optimization of Web Page Delivery

15 While a user peruses a given Web page, it would be possible to prefetch  
16 many of the pages to which his current page is currently linked. Then, when the  
17 user clicks a hyperlink, because his selection is already in the local buffer it can  
18 be delivered almost instantly. Obviously, this could be made more sophisticated,  
19 with probabilistic methods used to determine which pages are most likely to next  
20 be chosen by the user, and thus which are the most logical candidates for  
21 prefetching.

1           While the invention has been particularly shown and described with  
2 reference to a preferred embodiment, it will be understood by those skilled in the art  
3 that various changes in form and detail may be made therein without departing from  
4 the spirit and scope of the invention.

5

0635MH-40913 Patent Application